ATRIA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ASKB Campus, 1st Main Road, AG's Colony,
Anand Nagar, Bengaluru, Karnataka 560024
www.atria.edu

"MOBILE APPLICATION DEVELOPMENT WITH WEB TECHNOLOGY LAB MANUAL"

Prepared by

1. **Prof. SUHAS A BHYRATAE**
2. **Prof. SHRUTHI B**
3. **Prof. SYEDA ROSHNI AHMED**
4. **Prof. UZMA SULTHANA**
5. **Prof. PRAPULLA G**

Table of Contents

MOBILE APPLICATION DEVELOPMENT WITH WEB TECHNOLOGY

Android offers a variety of ways to present content to a user. To provide a user experience that's consistent with the rest of the platform, it's usually best to build a native app that incorporates framework-provided experiences, such as Android App Links or Search. Additionally, you can use Google Play-based experiences, such as App Actions and Slices, where Google Play services are available. Some apps, however, may need increased control over the UI.

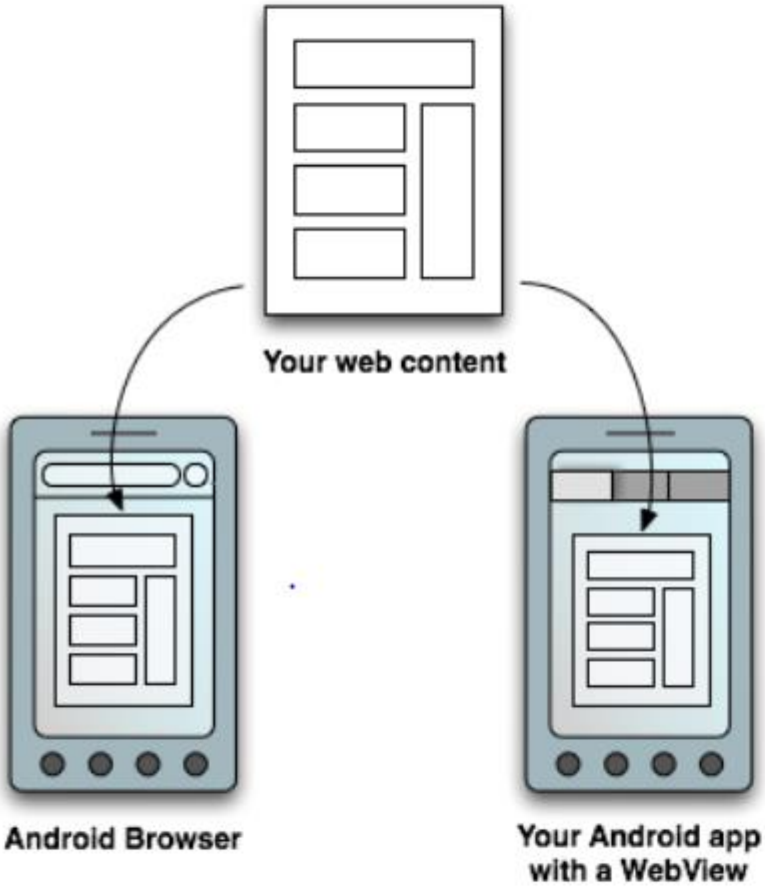


Fig.1 Web Content View via Android Application and Web Application

The above figure illustrates how you can provide access to your web pages from either a browser or your own Android app. The WebView framework allows you to

specify viewport and style properties that make your web pages appear at the proper size and scale on all screen configurations for all major web browsers. You can even define an interface between your Android app and your web pages that allows JavaScript in the web pages to call upon APIs in your app—providing Android APIs to your web-based application.
However, you shouldn't develop an Android app simply as a means to view your website. Rather, the web pages you embed in your app should be designed especially for that environment.

Programmes for showing the similarities between the WEB VIEWS and the Android Views have been shown below:

Program 1: Designing a Calculator using Web Concepts

Objective: Designing a simple calculator using HTML and JAVA SCRIPT to perform the following operations: sum, product, difference and quotient.

Input: Two integer numbers and select the operation to perform.
Output: Display Arithmetic Operations result.

Web View for designing a calculator using HTML and JAVASCRIPT: Here a Calculator is going to be created with HTML code.

- **Calculator Title**: This is the title at the top of our application, "CALCULATOR".
- **Output Screen**: This will be our output screen, where all text will be shown. Like the input that the user will type and the answer calculated from the user input. So, we can again break down this into two smaller pieces as shown below:
    - **Input** : This will be the input given by the user.
    - **Output**: This will be the result calculated from user input

Steps in Creating the Calculator using HTML and JAVASCRIPT:

1)Add TextView component & change the following properties:

```
border: 1px solid black;
 width: 33%;
 text-align: center;
```

background-color: DarkGray;
border-collapse:collapse;

2) Add PlainText(EditText) component & change the following properties in the script to perform the following operations:

   Enter the first number: Value 1

   Enter the second number: Value 2

   Final result by performing the operations of ADD,SUB,MUL,DIV

3) Add PlainText(EditText) component & change the following properties create a table for the calculator inside the body tag where the table tag is enclosed:

   Row span: to create rows in the calculator

   Col span: to create columns in the calculator

4) Add function component to accept the input numbers in the form of:

   Value1= Enter num1

   Value 2=Enter num2

   Value=Result

5) Add 4 Buttons & rename the four buttons "+", "-", "*" and "/" with following operations:

   Onclick: "Add"(+ Button)

   Onclick: "Sub"(- Button)

   Onclick: "Mul"(* Button)

   Onclick: "Div"(/ Button)

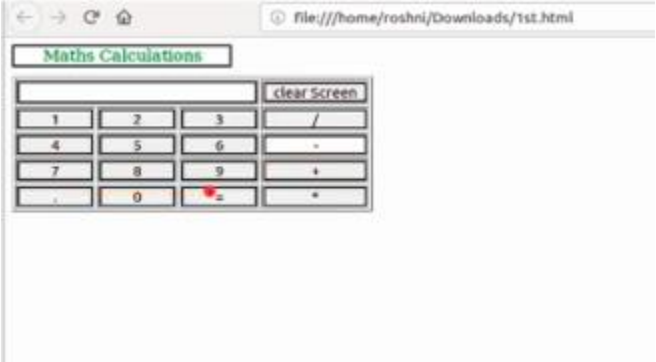
Code for designing a simple calculator using HTML and JAVASCRIPT to perform the following operations: sum, product, difference and quotient.

CODE:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
table, td, th
{
border: 1px solid black;
width: 33%;
text-align: center;
background-color: DarkGray;
border-collapse:collapse;
}
table { margin: auto; }
input { text-align:right; }
</style>
<script type="text/javascript">
function calc(clicked_id)
{
var val1 = parseFloat(document.getElementById("value1").value);
var val2 = parseFloat(document.getElementById("value2").value);
if(isNaN(val1)||isNaN(val2))
alert("ENTER VALID NUMBER");
else if(clicked_id=="add")

document.getElementById("answer").value=val1+val2;
else if(clicked_id=="sub")
document.getElementById("answer").value=val1-val2;
else if(clicked_id=="mul")
document.getElementById("answer").value=val1*val2;
else if(clicked_id=="div")
document.getElementById("answer").value=val1/val2;
}
function cls()
{
value1.value="0";
value2.value="0";
answer.value="";
}
</script>
</head>
<body>
<table>
<tr><th colspan="4"> SIMPLE CALCULATOR </th></tr>
<tr><td>value1</td><td><input type="text" id="value1"
value="0"/></td>
<td>value2</td><td><input type="text" id="value2" value="0"/>
</td></tr>
<tr><td><input type="button" value="Addition" id = "add"
onclick="calc(this.id)"/></td>
<td><input type="button" value="Subtraction" id = "sub"
```

```
onclick="calc(this.id)"/></td>
<td><input type="button" value="Multiplication" id = "mul"
onclick="calc(this.id)"/></td>
<td><input type="button" value="Division" id ="div"
onclick="calc(this.id)"/></td></tr>
<tr><td>Answer:</td><td> <input type="text" id="answer" value=""
disabled/></td>
<td colspan="2"><input type="button" value="CLEAR ALL"
onclick="cls()"/></td>
</tr>
</table>
</body>
</html>
```

OUTPUT:



file:///home/roshni/Downloads/1st.html

Calculator Application using Mobile Application Development

Objective: Designing a simple calculator using XML(Designing) and JAVA to perform the following operations: sum, product, difference and quotient.

Input: Two integer numbers and select the operation to perform.
Output: Display Arithmetic Operations result.

Steps for create Android View (Design) using xml file:

1) Firstly, Create an Application by Name “calciApplication”
2) Go to xml code of design change the layout to “RelativeLayout”
3) Add TextView component & change the following properties:
    - Size: 36sp
    - Text: Simple Calci
    - Center-Align
4) Add PlainText(EditText) component & change the following properties in XML Code:
    - Text: “”
    - Hint: “Enter the Number 1”
    - id: “@+id/editText1”
5) Add PlainText(EditText) component & change the following properties in XML Code:
    - Text: “”
    - Hint: “Enter the Number 2 ”
    - id: “@+id/editText2”
6) Add TextView component to display result & change the following properties:
    - Size: 40dp
    - Text: “0”
    - Center-Align
    - id: “@+id/textView1”
7) Add 4 Buttons & rename the four buttons “Add”, “Sub”,”Mul” and “div” with following addition:
    - Onclick: “doAdd”(Add Button)
    - Onclick: “doSub”(Sub Button)
    - Onclick: “doMul”(Mul Button)

    Onclick: “doDiv”(Div Button)

XML-CODE

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

            <TextView
        android:layout_width="209dp"
        android:layout_height="60dp"
```

```
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
   android:layout_alignParentBottom="true"
android:layout_marginEnd="108dp"
android:layout_marginRight="108dp"
android:layout_marginBottom="530dp"
   android:text="Simple Calci"
   android:textSize="36sp"
   app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

   <EditText
android:id="@+id/editText2"
   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="115dp"
android:layout_marginRight="115dp"
android:layout_marginBottom="364dp"
   android:ems="10"
   android:hint="Enter the Number 2"
android:inputType="textPersonName"
   android:text=""
android:textColorHighlight="#FFFFFF" />

   <EditText
android:id="@+id/editText1"
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="110dp"
android:layout_marginRight="110dp"
android:layout_marginBottom="439dp"
   android:ems="10"
   android:hint="Enter the Number 1"
   android:inputType="textPersonName"
```

```
    android:text=""
android:textColorHighlight="#FFFFFF" />

    <Button
android:id="@+id/button"
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="260dp"
android:layout_marginRight="260dp"
android:layout_marginBottom="175dp"
    android:text="ADD"
android:textStyle="bold"
    android:onClick="add"
app:backgroundTint="#E8F381" />

    <Button
android:id="@+id/button3"
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="266dp"
android:layout_marginRight="266dp"
android:layout_marginBottom="61dp"
    android:text="MUL"
    android:onClick="mul"
app:backgroundTint="#A1FAA4" />

    <Button
android:id="@+id/button4"
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="108dp"
android:layout_marginRight="108dp"
android:layout_marginBottom="63dp"
```

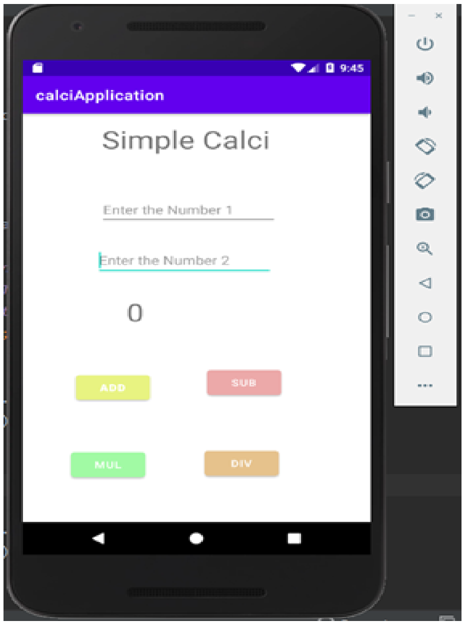
```
      android:text="DIV"
      android:onClick="div"
    app:backgroundTint="#E6C28C" />

      <Button
    android:id="@+id/button2"
    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="105dp"
    android:layout_marginRight="105dp"
    android:layout_marginBottom="182dp"
      android:text="SUB"
      android:onClick="sub"
    app:backgroundTint="#ECA9A9" />
      <TextView
      android:id="@+id/tv1"
    android:layout_width="86dp"
    android:layout_height="61dp"
    android:layout_alignParentEnd="true"
      android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="202dp"
    android:layout_marginRight="202dp"
    android:layout_marginBottom="274dp"
      android:text="0"
      android:textSize="36sp" />
</RelativeLayout>
```

Java code to perform Arithmetic Operations:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity {
EditText e1,e2;
TextView tv;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 =(EditText)findViewById(R.id.editText1);
        e2 = (EditText)findViewById(R.id.editText2);
        tv= (TextView)findViewById(R.id.tv1);
        }
        public void add(View v){
        int a1=Integer.parseInt(e1.getText().toString());
        int a2= Integer.parseInt(e2.getText().toString());
        int result=a1+a2;
        tv.setText(""+result);
        }
        public void sub(View v){
        int a1=Integer.parseInt(e1.getText().toString());
        int a2= Integer.parseInt(e2.getText().toString());
        int result=a1-a2;
           tv.setText(""+result);
        }
        public void mul(View v){
        int a1=Integer.parseInt(e1.getText().toString());
        int a2= Integer.parseInt(e2.getText().toString());
        int result=a1*a2;
        tv.setText(""+result);
        }
        public void div(View v){
        float a1=Integer.parseInt(e1.getText().toString());
        float a2= Integer.parseInt(e2.getText().toString());
        float result=a1/a2;
        tv.setText(""+result);
        }
}
```

OUTPUT:


9:45
calciApplication
Simple Calci
Enter the Number 1
Enter the Number 2
0
ADD
SUB
MUL
DIV

Programme 2: To create an web design for keeping track of visitors visiting the page.

Objective: Designing a web page to keep track of Visitors Visiting the Web Page

Input: PHP program to track the users visiting the web page

Output: Display the name of the user along with the count.

Steps in Creating the Visiting the Visitor activity using PHP as the scripting language:

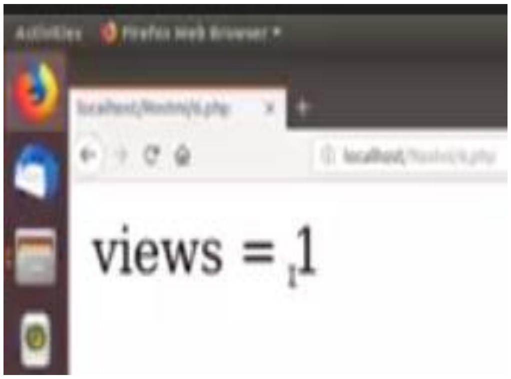
1. Counter.txt: This assigns the filename to a variable to be used throughout the rest of the script. In this case I used counter.txt. The first thing to do after implementing this script is to create counter.txt on your server.
2. open($name,"r"): it opens the file in read mode
3. fscanf($file,"%d"): The function fscanf() is similar to sscanf(), but it takes its input from a file associated with `stream` and interprets the input according to the specified `format`
4. $hits[0]++ : to either populate a new element or increment an existing element.

CODE :

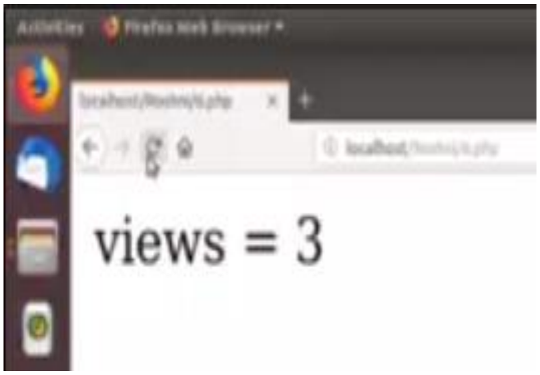
```
<?php
print "<h3> REFRESH PAGE </h3>";
$name="counter.txt";
$file = fopen($name,"r");
$hits= fscanf($file,"%d");
fclose($file);
$hits[0]++;
$file = fopen($name,"w");
fprintf($file,"%d",$hits[0]);
fclose($file);
print "The visitor Shreyansh has visited the site ".$hits[0] . "times" ;
```

```
?>
```

OUTPUT:



```
views = 1
```



```
views = 3
```

Create an application to design a Visiting Card using Mobile Application Development

Objective: Designing a Visiting card of an employee using XML(Designing) and JAVA

Input: Usage of components on activity like ImageView, TextView with different layouts,and entering the employee details which had to be displayed on the visiting card.

Output: Display an employee details on the screen in the visiting card layout.

Steps for create Android View (Design) using xml file:

1) Firstly, Create an Application by Name “VisitingCardApp”
2) Go to xml code of design change the layout to “RelativeLayout”
3) Add TextView component change the following properties:
    - Size: 38dp
    - Text: AIT
    - Align left top

4) Add ImageView to design and in type choose “IC_LAUNCHER_FOREGROUND”
   - Download the logo & copy the same in res->drawable folder
   -

   In xml code of imageview change srcCompat=”@drawable/logo”
   - Align right top

5) Add View component & change the following properties:
   - Height: 4dp
   - Background: “#4444” (black color)

6) Add TextView component change the following properties:
   - Size: 20dp
   - Text: Uzma Sulthana
   - Style: Bold
   - Align center

7) Add TextView component change the following properties:
   - Size: 24sp
   - Text: Assistant Professor-ISE
   - Align center

8) Add TextView component change the following properties:
   - Size: 24dp
   - Text: Address: ASKB Campus, Anandnagar, | Bangalore-560024
   - Align: center

9) Add TextView component change the following properties:
   - Size: 24sp
   - Text: Email-uzma.sulthana@atria.edu
   - Align: center

10) Add TextView component change the following properties:

- Size: 24sp
- Text: Phone-9108380566

XML-CODE

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

            android:background="#FFFFFF"

    tools:context=".MainActivity">


            <TextView

        android:id="@+id/textView4"

        android:layout_width="371dp"

        android:layout_height="wrap_content"

        android:layout_alignParentStart="true"

            android:layout_alignParentLeft="true"

        android:layout_alignParentEnd="true"

        android:layout_alignParentRight="true"

        android:layout_alignParentBottom="true"

        android:layout_marginStart="28dp"
```

```
android:layout_marginLeft="28dp"
     android:layout_marginEnd="12dp"
android:layout_marginRight="12dp"
android:layout_marginBottom="219dp"
   android:text="Address:ASKB Campus, Anandnagar, | Bangalore - 560 024"
android:textAlignment="center"
   android:textSize="24sp" />

   <TextView
android:id="@+id/textView5"
android:layout_width="250dp"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
   android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
android:layout_marginStart="87dp"
android:layout_marginLeft="87dp"
android:layout_marginEnd="73dp"
     android:layout_marginRight="73dp"
android:layout_marginBottom="157dp"
   android:text="Ph No: 9108380566"
```

```
android:textAlignment="center"
   android:textSize="24sp" />

   <TextView
android:id="@+id/textView6"
   android:layout_width="367dp"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
android:layout_marginStart="25dp"
android:layout_marginLeft="25dp"
android:layout_marginEnd="19dp"
android:layout_marginRight="19dp"
android:layout_marginBottom="64dp"
   android:text="Email Id: uzma.sulthana@atria.edu"
android:textAlignment="center"
   android:textSize="24sp" />

   <TextView
android:id="@+id/textView3"
```

```
android:layout_width="367dp"
    android:layout_height="66dp"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
    android:layout_marginStart="32dp"
android:layout_marginLeft="32dp"
android:layout_marginEnd="12dp"
android:layout_marginRight="12dp"
android:layout_marginBottom="287dp"
    android:text="Assistant Professor-ISE"
    android:textAlignment="center"
    android:textSize="24sp" />

    <ImageView
android:id="@+id/imageView3"
android:layout_width="155dp"
android:layout_height="98dp"
android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
```

```
android:layout_marginEnd="12dp"

android:layout_marginRight="12dp"

android:layout_marginBottom="495dp"

app:srcCompat="@drawable/aitlogo" />


    <View

    android:id="@+id/view"

android:layout_width="wrap_content"

android:layout_height="4dp"

android:layout_alignParentBottom="true"

android:layout_marginBottom="487dp"

android:background="#4444" />


    <TextView

    android:id="@+id/textView2"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:layout_alignParentStart="true"

android:layout_alignParentLeft="true"

android:layout_alignParentEnd="true"

    android:layout_alignParentRight="true"

android:layout_alignParentBottom="true"

android:layout_marginStart="95dp"
```

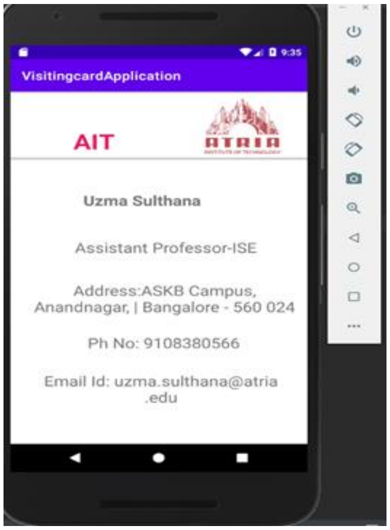
```
android:layout_marginLeft="95dp"

android:layout_marginEnd="140dp"

android:layout_marginRight="140dp"

        android:layout_marginBottom="401dp"

    android:text="Uzma Sulthana"

android:textAlignment="center"

    android:textSize="24sp"

android:textStyle="bold" />


    <TextView

android:id="@+id/textView7"

    android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignParentBottom="true"

android:layout_marginEnd="99dp"

android:layout_marginRight="99dp"

android:layout_marginBottom="495dp"

    android:layout_toStartOf="@+id/imageView3"

android:layout_toLeftOf="@+id/imageView3"

    android:text="AIT"

android:textColor="#E91E63"

    android:textSize="36sp"

android:textStyle="bold" />
```

```
</RelativeLayout>
```

JAVA-CODE

```
 package com.example.visitingcardapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

            protected void onCreate(Bundle savedInstanceState) {

             super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

            }

}
```

OUTPUT:



VisitingcardApplication

AIT

Uzma Sulthana

Assistant Professor-ISE

Address:ASKB Campus,
Anandnagar, | Bangalore - 560 024

Ph No: 9108380566

Email Id: uzma.sulthana@atria
.edu

Programme 3: Creating a web Page to show the SIGN UP ACTIVITY.

Objective: Designing a web page to have a login contents with username and password with submit action.
Input: To enter username and password with action
Output: Login to the web page content.

HTML Code:
Style.css:

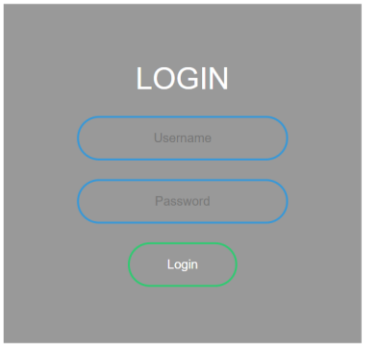
```
body{
  margin: 0;
  padding: 0;
  font-family: sans-serif;
  background: url(back.jpg);
 background-size: cover;
}
.box{
  width: 300px;
  padding: 40px;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%,-50%);
  background: rgba(0,0,0,0.4);
  text-align: center;
}
.box h1{
  color: white;
  text-transform: uppercase;
  font-weight: 500;
}
.box input[type = "text"],.box input[type = "password"]{
  border:0;
  background: none;
  display: block;
  margin: 20px auto;
  text-align: center;
  border: 2px solid #3498db;
  padding: 14px 10px;
  width: 200px;
  outline: none;
  color: white;
  border-radius: 24px;
  transition: 0.25s;
}
.box input[type = "text"]:focus,.box input[type = "password"]:focus{
  width: 280px;
  border-color: #2ecc71;
}
.box input[type = "submit"]{
  border:0;
  background: none;
  display: block;
  margin: 20px auto;
  text-align: center;
  border: 2px solid #2ecc71;
  padding: 14px 40px;
  outline: none;
  color: white;
  border-radius: 24px;
  transition: 0.25s;
  cursor: pointer;
}
.box input[type = "submit"]:hover{
  background: #2ecc71;
}
```

Index.html:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Login Form</title>
    <link rel="stylesheet" href="style.css">
  </head>
```

```
 <body>

<form class="box" action="index.html" method="POST">
 <h1>Login</h1>
 <input type="text" name="" placeholder="Username">
 <input type="password" name="" placeholder="Password">
 <input type="submit" name="" value="Login">
</form>
<script>
            var username=document.getElementById("Username").value;
            var password=document.getElementById("Password").value;
            if(username=="admin" && password="user")
                        {
                                    alert("Login Successfully ");
                                    return false;

                        }
else{
 alert("Login Failed....")
}
</script>
 </body>
</html>
```

OUTPUT:



Creating a web Page to show the SIGN UP ACTIVITY using android application development.

Objective: Designing a SIGNUp activity with Username and Password using XML(Designing) and writing JAVA code for validation of the password

Input: Enter details for signup activity, and validated password to the login page
Output: Display signup login pages, and need to login the activity page the validated password.

Steps for create Android View (Design) using xml file:

1) Firstly Create an Application by Name "SignUpApplication"
2) Go to xml code of design change the layout to "RelativeLayout"
3) Add TextView component & change the following properties:
   - Size: 24sp
   - Text: "Sign Up"
   - Center-Align
4) Add Email (EditText) component & change the following properties in XML Code:
   - Hint: "Email-ID"
   - id: "@+id/emailEditText"
5) Add Password (EditText) component & change the following properties in XML Code:
   - Hint: "Password"
   - id: "@+id/passwordEditText"
6) Add Button component & change the following properties in XML
   - Id: "@+id/signBtn"
   - Text: "Sign Up"

XML-CODE

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

```
android: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: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" />

 <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" />

 <EditText
android:id="@+id/SignUp_Password"
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="101dp"
```

```
        android:layout_marginRight="101dp"
          android:layout_marginBottom="385dp"
          android:ems="10"
          android:hint="Password"
        android:inputType="textPassword" />

  </RelativeLayout>
```

JAVA-CODE

```
package com.example.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;


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,loginActivity.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 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;

        }

}
```

1) Right click on Java folder-> new-> activity->empty activity-> name it as

“LoginActivity”

2) Go to xml code of design change the layout to “RelativeLayout”
3) Add TextView component & change the following properties:
   - Size: 38dp
   - Text: “Login”
   - Center-Align
4) Add Email (EditText) component & change the following properties in XML Code:
   - Hint: “Email ID”
   - id: “@+id/emailEditText”
5) Add Password (EditText) component & change the following properties in XML Code:
   - Hint: “Password”
   - id: “@+id/passwordEditText”
6) Add Button component & change the following properties in XML
   - Id: “@+id/loginBtn”
   - Text: “Login”

XML-CODE

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

   xmlns:app="http://schemas.android.com/apk/res-auto"

   xmlns:tools="http://schemas.android.com/tools"

   android: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: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: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>
```

JAVA-CODE

```
package com.example.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_login2);
```

```
    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,loginsuccessActivity.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);
```

```
                        }

                }

                });

                }

        }
```

1) Right click on Java folder-> new-> activity->empty activity-> name it as "LoginSuccessful"
2) Go to xml code of design change the layout to "RelativeLayout"
3) Add TextView component & change the following properties:
   - Size: 38dp
   - Text: "Login Successful"
   - Center-Align

XML-CODE

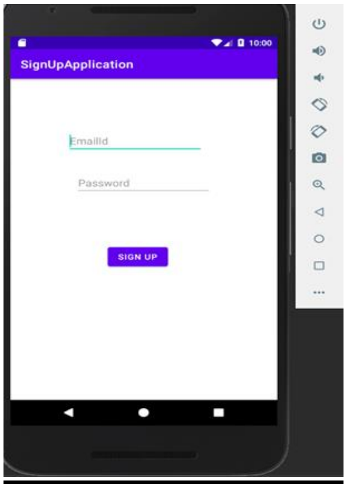
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
         android:layout_height="match_parent"
   tools:context=".loginsuccessActivity">
         <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="Login Successful"
         android:textSize="36sp"
      android:textStyle="bold" />
```

```
</RelativeLayout>
```

JAVA-CODE

```
package com.example.signupapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class loginsuccessActivity extends AppCompatActivity {

        @Override

        protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_loginsuccess);

        }}
```

OUTPUT:



Programme 4: Display the counter value in a TextViewcontrol using PHP

Objective: Designing a web page in order to keep the track of time using the counter value in the form of hours, minutes and seconds

Input: To keep the track of time in terms of hours, minutes and seconds
Output: Display number of counts on the screen by showing the time

Steps in Creating the counter value in the form of hours, minutes and seconds:

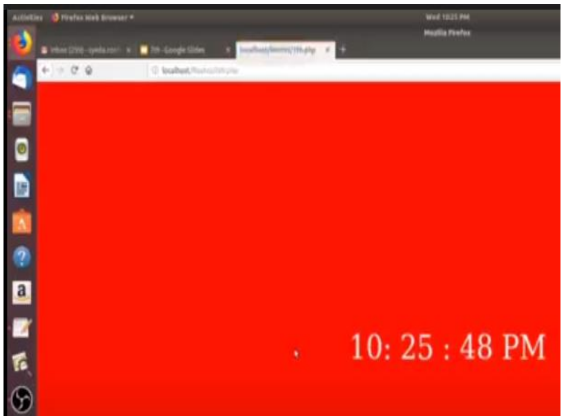
1. Designing the web Page with different background colour
2. Styling the concepts of the same and placing the text at an position and adding different properties.
3. Using PHP to accept the time in terms of counter Value

CODE:

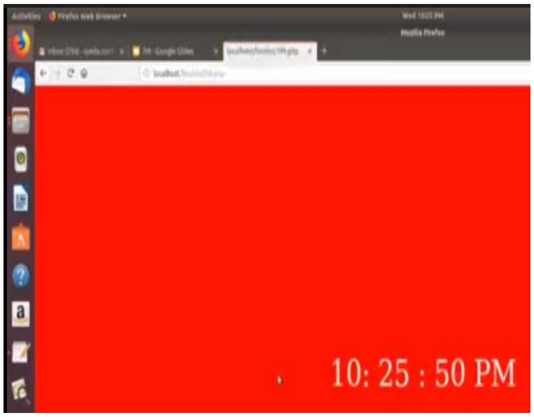
```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="refresh" content="1"/>
<style>
p {
color:white;
font-size:90px;
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
}
body{background-color:red;}
</style>
```

```
<p> <?php echo date(" h: i : s A");?> </p>
</head>
</html>
```

OUTPUT:



10: 25 : 48 PM



10: 25 : 50 PM

Display the counter value in a TextViewcontrol using time for MAD

Objective: Designing START & STOP button & text view using XML(Designing) and writing JAVA code for counter increment

Input: Press start & stop buttons
Output: Display number of counts on the screen

Steps for create Android View (Design) using xml file:

1) Firstly Create an Application by Name "CounterActivity"
2) Go to xml code of design change the layout to "RelativeLayout"
3) Add TextView component & change the following properties:
   - Size: 38dp
   - Text: "Counter Application"
   - Center-Align
4) Add TextView component & change the following properties:

- Text: “Counter Value”

5) Add Button components & change the following properties:
   - Size: 38dp
   - Text: Start
   - id: “@+id/btn_start”
6) Add Button components & change the following properties:
   - Size: 38dp
   - Text: Stop
   - id: “@+id/btn_stop”

XML-CODE

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

   xmlns:app="http://schemas.android.com/apk/res-auto"

         xmlns:tools="http://schemas.android.com/tools"

   android:layout_width="match_parent"

   android:layout_height="match_parent"

   tools:context=".MainActivity">


         <TextView

      android:id="@+id/textView1"

      android:layout_width="332dp"

         android:layout_height="wrap_content"

      android:layout_alignParentStart="true"

      android:layout_alignParentLeft="true"

      android:layout_alignParentEnd="true"
```

```
android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
android:layout_marginStart="41dp"
android:layout_marginLeft="41dp"
android:layout_marginEnd="38dp"
android:layout_marginRight="38dp"
android:layout_marginBottom="516dp"
    android:text="Counter Application"
    android:textSize="36sp"
android:textStyle="bold" />


    <Button
android:id="@+id/button1"
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="236dp"
android:layout_marginRight="236dp"
android:layout_marginBottom="89dp"
    android:text="Start"
       android:textSize="30sp"
```

```
app:backgroundTint="#4CAF50" />


    <Button
android:id="@+id/button2"
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="55dp"
android:layout_marginRight="55dp"
android:layout_marginBottom="92dp"
    android:text="STOP"
    android:textSize="30sp"
app:backgroundTint="#EC5449" />


    <TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
```

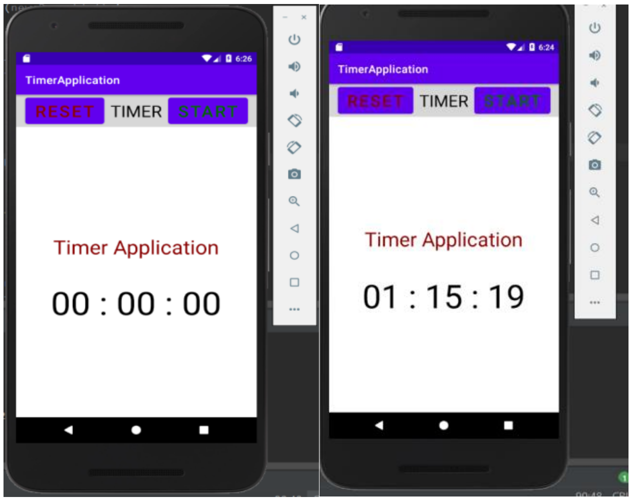
```
        android:layout_marginEnd="165dp"

        android:layout_marginRight="165dp"

        android:layout_marginBottom="434dp"

           android:text="counter value"

           android:textSize="18sp"

        android:textStyle="bold" />

</RelativeLayout>
```

JAVA-CODE

```
package com.example.counterapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;



public class MainActivity extends AppCompatActivity {

           TextView txtCounter;

           Button btn_start,btn_stop;

           int count=0;
```

```
    Handler customHandler=new Handler();

    @Override

    protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

    txtCounter= (TextView)findViewById(R.id.textView2);

    btn_start =(Button)findViewById(R.id.button1);

btn_stop=(Button)findViewById(R.id.button2);

    btn_start.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

    customHandler.postDelayed(updateTimerThread,0);

    }

    });

    btn_stop.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        customHandler.removeCallbacks(updateTimerThread);
    }

    });

      }

    private final Runnable updateTimerThread =new Runnable() {

    @Override

    public void run() {

        txtCounter.setText(""+count);

        customHandler.postDelayed(this,1000);

                count++;

          }

    }

    }
```

OUTPUT:



Programme 5: Web Designing page for showing the text growing and shrinking using web Java Script concept

Objective: Designing a web page in order to show the text growing ad text shrinking concept by increasing and decreasing the font size of the same.

Input: To create functions to show the concepts of text growing amd shrinking
Output: A play where text grows until it reaches 50 ms and shrinks by reducing the size of it.

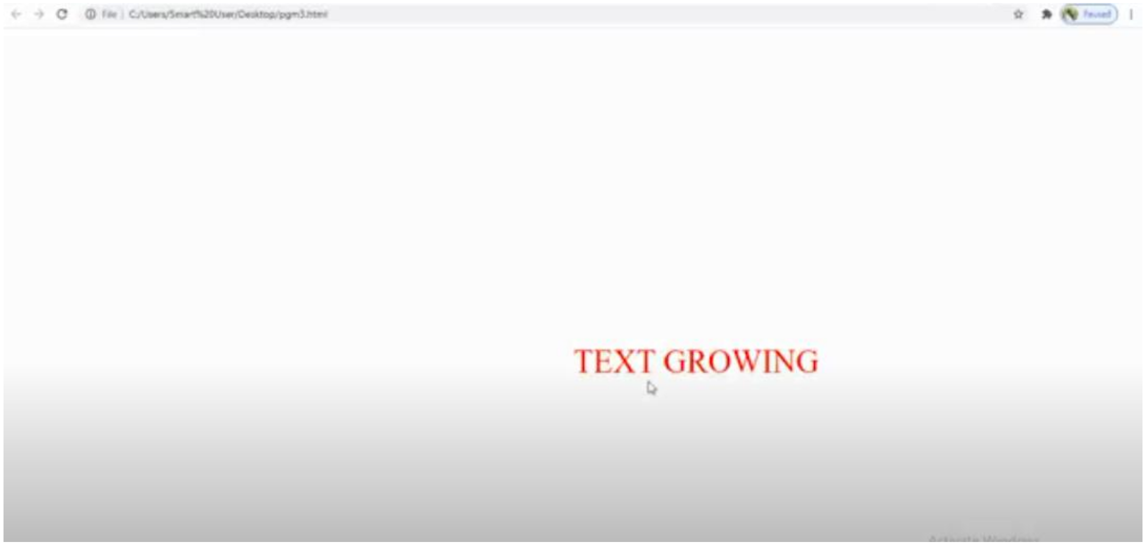
Steps in showing the concept of text growing and text shrinking.

1. Creating the web page by designing the position of the text that has to be placed at the centre and then finding the exact position to place the same using the ID attribute.
2. Setting the time interval for growing and shrinking of text using the function timer.

CODE:

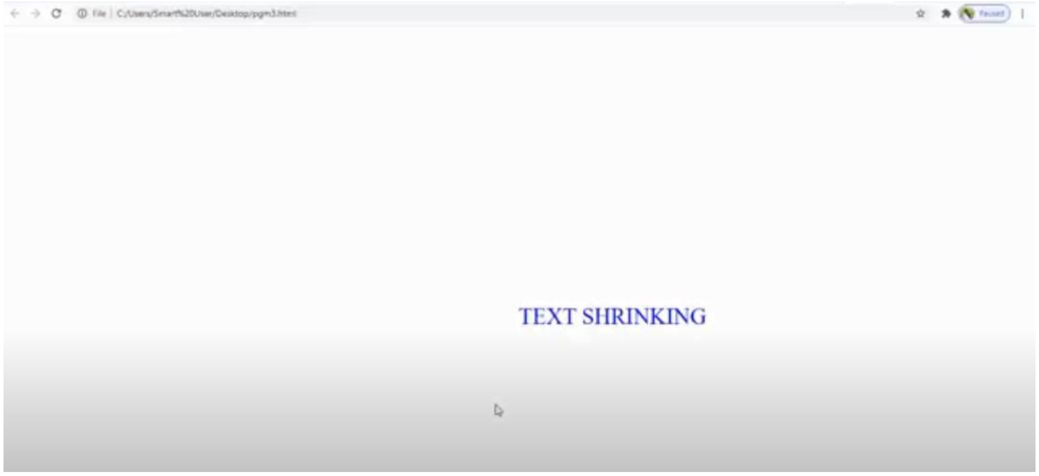
```
<!DOCTYPE HTML>
<html>
<head>
<style>
p {
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
}
</style>
</head>
<body>
<p id="demo"></p>
<script>
var var1 = setInterval(inTimer, 1000);
var fs = 5;
var ids = document.getElementById("demo");
function inTimer() {
ids.innerHTML = 'TEXT GROWING';
ids.setAttribute('style', "font-size: " + fs + "px; color:
red");
fs += 5;
if(fs >= 50 ){
clearInterval(var1);
var2 = setInterval(deTimer, 1000);
}
}
function deTimer() {
fs -= 5;
ids.innerHTML = 'TEXT SHRINKING';
ids.setAttribute('style', "font-size: " + fs + "px; color:
blue");
if(fs === 5 ){
clearInterval(var2);
}
}
</script>
</body>
</html>
```

Output:



C:/Users/Smart%20User/Desktop/pgm3.html

TEXT GROWING



C:/Users/Smart%20User/Desktop/pgm3.html

TEXT SHRINKING

Mobile Application Development programme to Convert text to Speech

Objective: Designing EDITTEXT using XML(Designing) and writing JAVA code to convert text to speech

Input: entering text in the text view
Output: Converting text to speech

Steps for create Android View (Design) using xml file:

1) Firstly Create an Application by Name “TextToSpeech”
2) Go to xml code of design change the layout to “RelativeLayout”
3) Add TextView component & change the following properties:
4) Size: 38dp
5) Text: Text2Speech App
6) Center-Align
7) Add PlainText(EditText) component & change the following properties in XML Code:
   - Text: “”
   - Hint: “Enter the text to be converted”
   - id: “@+id/editText”
8) Add Button component & change the following properties in XML Code:
   - Name: Convert
   - onClick: convert

XML-CODE:

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

   xmlns:app="http://schemas.android.com/apk/res-auto"

         xmlns:tools="http://schemas.android.com/tools"

   android:layout_width="match_parent"

   android:layout_height="match_parent"
```

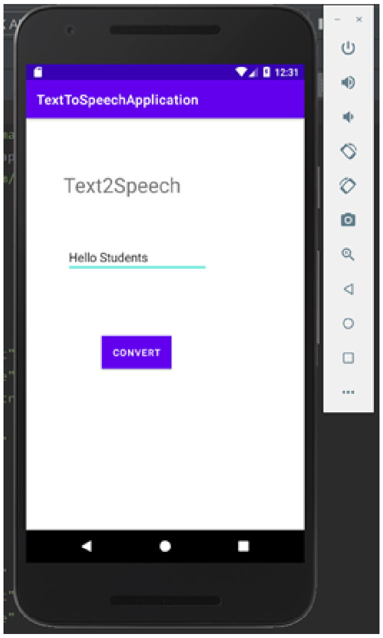
```
tools:context=".MainActivity">

    <TextView
  android:id="@+id/textView"
  android:layout_width="335dp"
    android:layout_height="wrap_content"
  android:layout_alignParentEnd="true"
  android:layout_alignParentBottom="true"
  android:layout_marginEnd="21dp"
  android:layout_marginBottom="486dp"
  android:text="Text2Speech"
    android:textSize="30sp" />
    <EditText
  android:id="@+id/editText"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignParentEnd="true"
  android:layout_alignParentBottom="true"
      android:layout_marginEnd="142dp"
  android:layout_marginBottom="377dp"
    android:ems="10"
    android:hint="Enter text here"
  android:inputType="textPersonName" />

    <Button
```

```
        android:id="@+id/convert"
            android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="196dp"
        android:layout_marginBottom="236dp"
            android:onClick="convert"
        android:background="#6CEC71"
            android:text="CONVERT" />
</RelativeLayout>
```

JAVA-CODE:

```
package com.example.texttospeechapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.EditText;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
        EditText e1;
        TextToSpeech t1;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1=findViewById(R.id.editText);
        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
        public void onInit(int status) {
        if(status!=TextToSpeech.ERROR){
              t1.setLanguage(Locale.UK);
        }
        }
        });
        }
        public void convert(View V){
        String tospeak=e1.getText().toString();
       t1.speak(tospeak,TextToSpeech.QUEUE_FLUSH,null);
        }
}
```

OUTPUT:



TextToSpeechApplication

Text2Speech

Hello Students