

1. Cordova - alert messages:

Index.html:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Alert App</title>
</head>
<body>
  <h1>Hello Cordova!</h1>
  <button onclick="showAlert()">Show Alert</button>

  <script>
    function showAlert() {
      alert("Hello from Cordova!");
    }
  </script>
</body>
</html>
```

2. React - text colour change

```
import React, { useState } from 'react';

const ColorChanger = () => {
  const [textColor, setTextColor] = useState('black');

  const changeColor = () => {
    // Generate a random color for demonstration purposes
    const randomColor = `#${Math.floor(Math.random()*16777215).toString(16)}`;
    setTextColor(randomColor);
  };

  return (
    <div>
      <p style={{ color: textColor }}>
        This is some text with dynamic color!
      </p>
      <button onClick={changeColor}>
        Change Color
      </button>
    </div>
  );
}
```

```

    </div>
  );
};

export default ColorChanger;

```

3. React native - alignment change

```

import React, { useState } from 'react';
import { View, Text, StyleSheet, Button } from 'react-native';

const TextAlignmentExample = () => {
  const [alignment, setAlignment] = useState('center');

  const changeAlignment = (newAlignment) => {
    setAlignment(newAlignment);
  };

  return (
    <View style={styles.container}>
      <Text style={[styles.text, { textAlign: alignment }]}>
        This is a dynamically aligned text.
      </Text>
      <View style={styles.buttonContainer}>
        <Button
          title="Left Align"
          onPress={() => changeAlignment('left')}
        />
        <Button
          title="Center Align"
          onPress={() => changeAlignment('center')}
        />
        <Button
          title="Right Align"
          onPress={() => changeAlignment('right')}
        />
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,

```

```

    justifyContent: 'center',
    alignItems: 'center',
  },
  text: {
    fontSize: 20,
    marginBottom: 20,
  },
  buttonContainer: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    width: '60%', // Adjust the width according to your layout
  },
});

export default TextAlignmentExample;

```

4. React - Login page

```

import React, { useState } from 'react';
import { View, Text, StyleSheet, TextInput, Button, Alert } from 'react-native';

const LoginPage = () => {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  const handleLogin = () => {
    // Simple validation
    if (username === 'user' && password === 'password') {
      Alert.alert('Login Successful', 'Welcome back, User!');
    } else {
      Alert.alert('Login Failed', 'Invalid username or password. Please try again.');
```

```

        style={styles.input}
        placeholder="Password"
        secureTextEntry
        onChangeText={({text) => setPassword(text)}
        value={password}
      />
      <Button
        title="Login"
        onPress={handleLogin}
      />
    </View>
  );
};

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    padding: 16,
  },
  title: {
    fontSize: 24,
    marginBottom: 16,
  },
  input: {
    height: 40,
    borderColor: 'gray',
    borderWidth: 1,
    marginBottom: 16,
    padding: 8,
    width: '100%',
  },
});

```

```
export default LoginPage;
```

5. React - page navigation

```
npm install @react-navigation/native @react-navigation/stack
```

```

// App.js
import React from 'react';
import { NavigationContainer } from '@react-navigation/native';

```

```

import { createStackNavigator } from '@react-navigation/stack';
import HomePage from './components/HomePage';
import NextPage from './components/NextPage';

const Stack = createStackNavigator();

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home">
        <Stack.Screen name="Home" component={HomePage} />
        <Stack.Screen name="NextPage" component={NextPage} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

export default App;

// components/HomePage.js
import React from 'react';
import { View, Text, Button, StyleSheet } from 'react-native';

const HomePage = ({ navigation }) => {
  const navigateToNextPage = () => {
    navigation.navigate('NextPage');
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Home Page</Text>
      <Button
        title="Go to Next Page"
        onPress={navigateToNextPage}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
});

```

```
},  
  title: {  
    fontSize: 24,  
    marginBottom: 16,  
  },  
});
```

```
export default HomePage;
```

```
// components/NextPage.js  
import React from 'react';  
import { View, Text, StyleSheet } from 'react-native';
```

```
const NextPage = () => {  
  return (  
    <View style={styles.container}>  
      <Text style={styles.title}>Next Page</Text>  
    </View>  
  );  
};
```

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    justifyContent: 'center',  
    alignItems: 'center',  
  },  
  title: {  
    fontSize: 24,  
    marginBottom: 16,  
  },  
});
```

```
export default NextPage;
```

6. Android - Text colour

```
//java
import android.graphics.Color;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView textView;

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

        // Find the TextView by its ID
        textView = findViewById(R.id.textView);

        // Find the Button by its ID
        Button changeColorButton = findViewById(R.id.changeColorButton);

        // Set an OnClickListener for the button
        changeColorButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Change the text color to red
                textView.setTextColor(Color.RED);
            }
        });
    }
}

//xml
<!-- res/layout/activity_main.xml -->

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
android:paddingLeft="16dp"
android:paddingTop="16dp"
android:paddingRight="16dp"
android:paddingBottom="16dp"
tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, World!"
    android:textSize="20sp"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="20dp"/>
```

```
<Button
    android:id="@+id/changeColorButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Change Color"
    android:layout_below="@id/textView"
    android:layout_centerHorizontal="true"/>
```

```
</RelativeLayout>
```

7. Android - alert message

```
//java
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        Button showAlertButton = findViewById(R.id.showAlertButton);

        showAlertButton.setOnClickListener(new View.OnClickListener() {
            @Override
```



```

        public void onClick(View v) {
            // Display a short toast message
            Toast.makeText(MainActivity.this, "Alert: This is a text alert!",
                Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

```

//xml
<Button
    android:id="@+id/showAlertButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Alert"
    android:layout_below="@id/textView"
    android:layout_centerHorizontal="true"/>

```

8. Android- layout based form

```

//java
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        final EditText editName = findViewById(R.id.editName);
        final EditText editEmail = findViewById(R.id.editEmail);
        Button showInfoButton = findViewById(R.id.showInfoButton);

        showInfoButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = editName.getText().toString();
                String email = editEmail.getText().toString();
            }
        });
    }
}

```

```

        String message = "Name: " + name + "\nEmail: " + email;

        // Display a Toast message with entered information
        Toast.makeText(MainActivity.this, message, Toast.LENGTH_SHORT).show();
    }
    });
}
}
}

```

```
//xml
```

```
<!-- res/layout/activity_main.xml -->
```

```
<LinearLayout
```

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

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

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

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

```
    android:padding="16dp"
```

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

```
    <EditText
```

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

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

```
        android:layout_height="wrap_content"
```

```
        android:hint="Enter your name"/>
```

```
    <EditText
```

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

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

```
        android:layout_height="wrap_content"
```

```
        android:inputType="textEmailAddress"
```

```
        android:hint="Enter your email"/>
```

```
    <Button
```

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

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="Show Info"/>
```

```
</LinearLayout>
```

