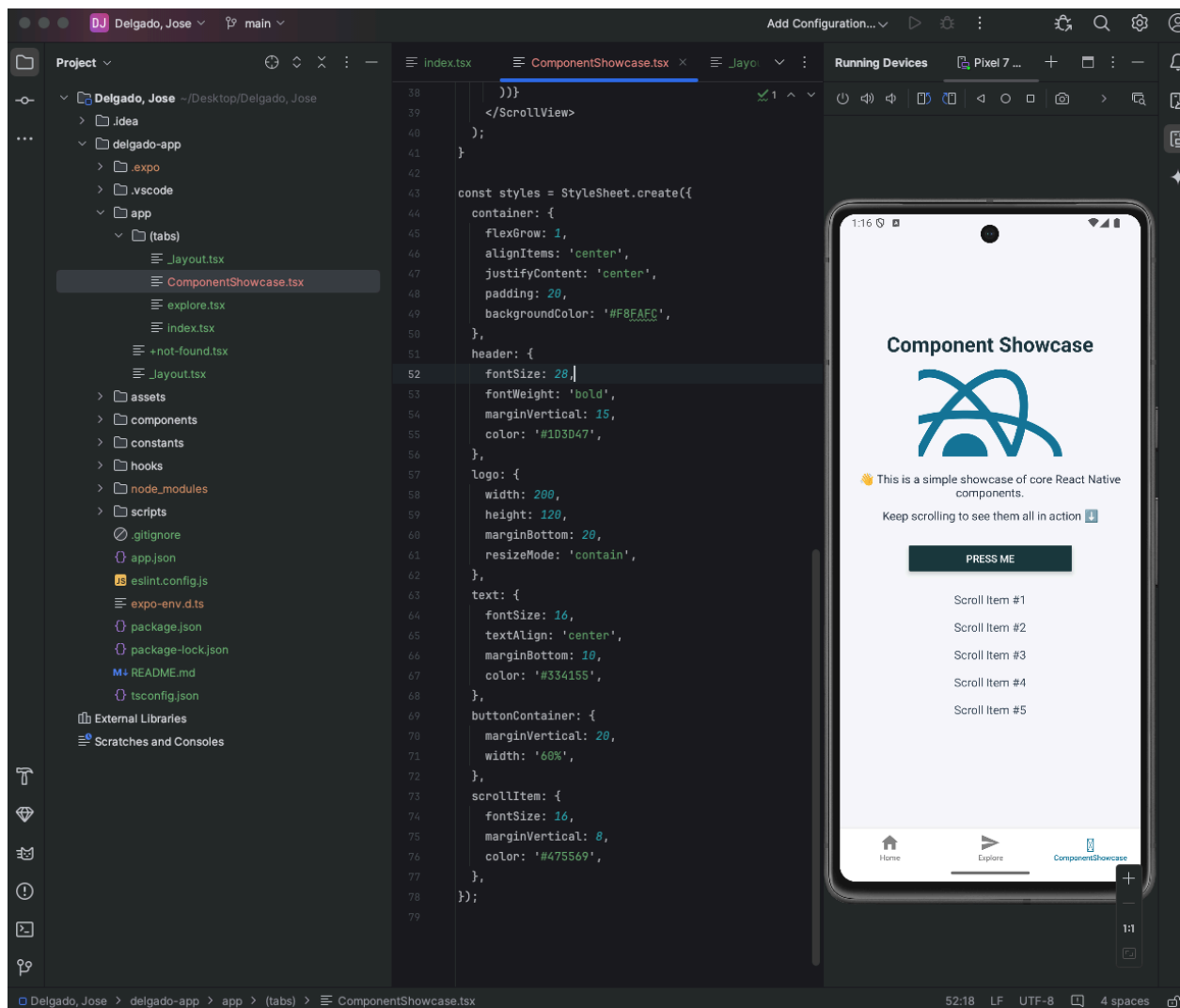


Delgado, Jose Rafael Achilles L.

## Activity 2: Component Scavenger Hunt

### 1. Create a new screen called ComponentShowcase



### 2. Use the following components: Text, Button, Image, ScrollView.

Text Component:

TypeScript

```
<Text style={[styles.header, isTablet && styles.headerTablet]}>
  Component Showcase
</Text>
```

```
<Text style={[styles.text, isTablet && styles.textTablet]}>
  Welcome to my component showcase page.{`\n`}
  This displays: Text, Button, Image, and ScrollView!
</Text>
```

### Button Component:

```
TypeScript
<Button
    title="Test Button"
    color="#38bdf8"
    onPress={() =>
        Alert.alert('Button Pressed!', 'You pressed the button!')
    }
/>
```

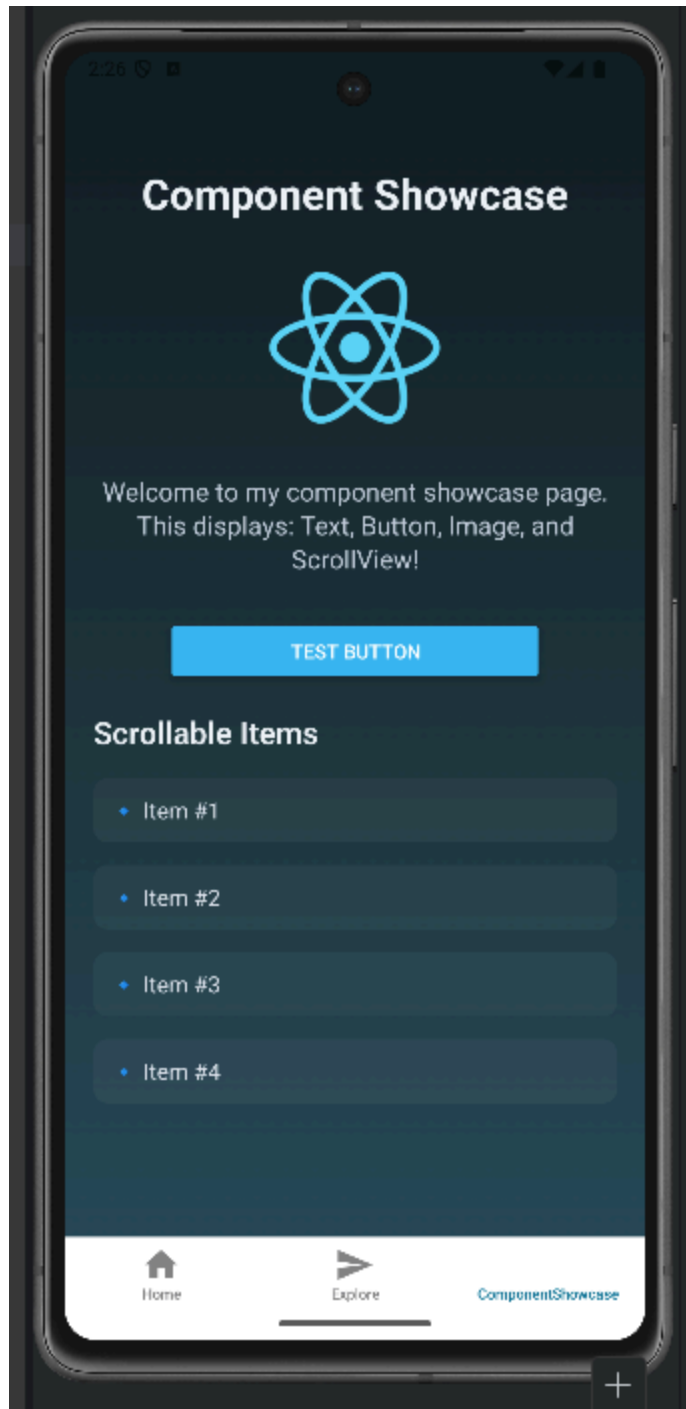
### Image Component:

```
TypeScript
<Image
    source={require('@assets/images/react-logo.png')}
    style={[styles.logo, isTablet && styles.logoTablet]}
/>
```

### ScrollView Component:

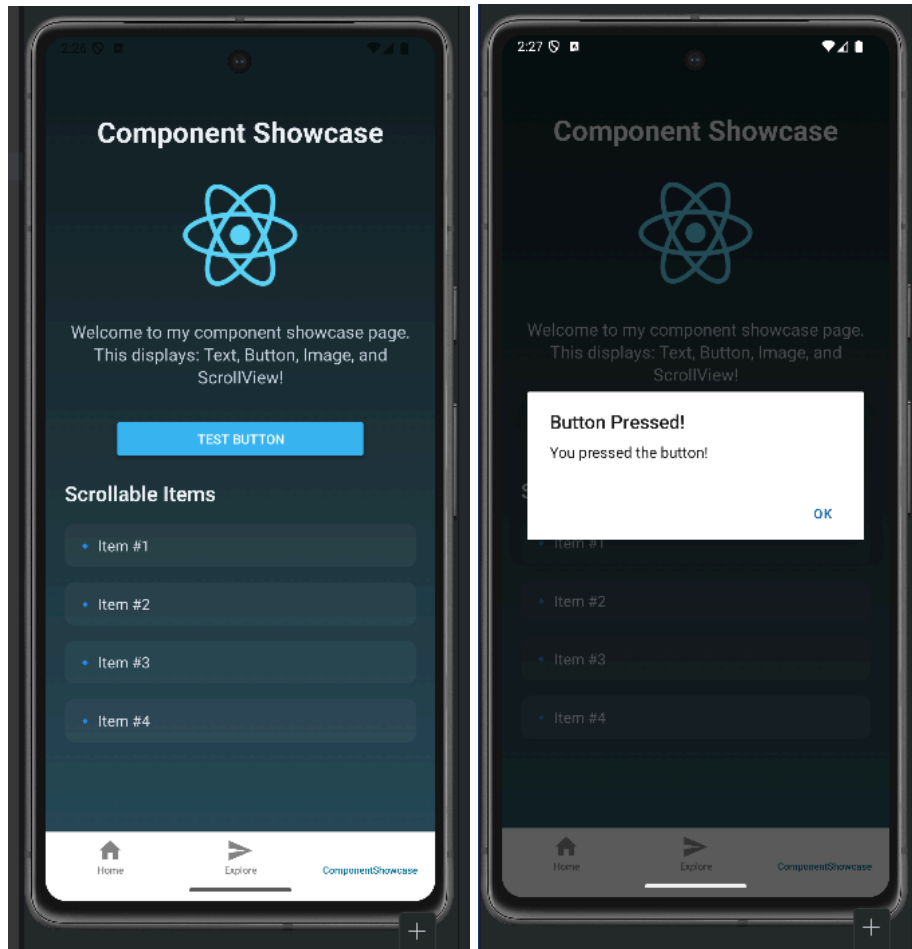
```
TypeScript
<ScrollView
    style={[styles.scrollContainer, isTablet &&
styles.scrollContainerTablet]}
    contentContainerStyle={[styles.scrollContent, isTablet &&
styles.scrollContentTablet]}
>
    {[...Array(20)].map((_, i) => (
        <View key={i} style={[styles.scrollItemBox, isTablet &&
styles.scrollItemBoxTablet]}>
            <Text style={[styles.scrollItem, isTablet &&
styles.scrollItemTablet]}>
                ♦ Item #{i + 1}
            </Text>
        </View>
    ))}
</ScrollView>
```

3. Style the layout to be clean and readable. (Tab and Phone)



Phone View

4. Run the app and ensure all components are visible and functional.



Before and After pressing “Test Button”

5. Submit the code and a screenshot of the running app.

### ComponentShowcase.tsx

```
TypeScript
import React from 'react';
import { StyleSheet, Text, View, Button, Image, ScrollView, Alert,
useWindowDimensions } from 'react-native';
import { LinearGradient } from 'expo-linear-gradient';

export default function ComponentShowcase() {
  const { width } = useWindowDimensions();
  const isTablet = width >= 800; // Pixel Tablet breakpoint

  return (
    <LinearGradient
```

```

    colors={['#0f2027', '#203a43', '#2c5364']}
    style={styles.gradientBackground}
  >
    <View style={[styles.container, isTablet && styles.tabletContainer]}>

      {/** Left Section: Info */}
      <View style={[styles.infoSection, isTablet &&
styles.infoSectionTablet]}>
        <Text style={[styles.header, isTablet && styles.headerTablet]}>
          Component Showcase
        </Text>

        <Image
          source={require('@assets/images/react-logo.png')}
          style={[styles.logo, isTablet && styles.logoTablet]}
        />

        <Text style={[styles.text, isTablet && styles.textTablet]}>
          Welcome to my component showcase page.{`\n`}
          This displays: Text, Button, Image, and ScrollView!
        </Text>

        <View style={[styles.buttonContainer, isTablet &&
styles.buttonContainerTablet]}>
          <Button
            title="Test Button"
            color="#38bdf8"
            onPress={() =>
              Alert.alert('Button Pressed!', 'You pressed the button!')
            }
          />
        </View>
      </View>

      {/** Right Section: Scrollable List */}
      <View style={[styles.scrollSection, isTablet &&
styles.scrollSectionTablet]}>
        <Text style={[styles.subHeader, isTablet && styles.subHeaderTablet]}>
          Scrollable Items
        </Text>
        <ScrollView
          style={[styles.scrollContainer, isTablet &&
styles.scrollContainerTablet]}

```

```

        contentContainerStyle=[styles.scrollContent, isTablet &&
styles.scrollContentTablet]]
    >
        {[...Array(20)].map((_, i) => (
            <View key={i} style=[styles.scrollItemBox, isTablet &&
styles.scrollItemBoxTablet]]>
                <Text style=[styles.scrollItem, isTablet &&
styles.scrollItemTablet]]>
                    ♦ Item #{i + 1}
                </Text>
            </View>
        )]}
    </ScrollView>
</View>
</View>
</LinearGradient>
);
}

```

```

const styles = StyleSheet.create({
  gradientBackground: {
    flex: 1,
    paddingTop: 40,
  },
  container: {
    flex: 1,
    alignItems: 'center',
    padding: 20,
  },
  tabletContainer: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'flex-start', // top align
    paddingHorizontal: 40,
  },
  infoSection: {
    width: '100%',
    alignItems: 'center',
  },
  infoSectionTablet: {
    flex: 1,
    alignItems: 'flex-start',
    marginRight: 30,
  },
},

```

```
scrollSection: {
  width: '100%',
},
scrollSectionTablet: {
  flex: 1,
  height: '100%', // take full height
},
header: {
  fontSize: 30,
  fontWeight: 'bold',
  marginVertical: 20,
  color: '#f1f5f9',
  textAlign: 'center',
},
headerTablet: {
  fontSize: 40,
  textAlign: 'left',
},
logo: {
  width: 220,
  height: 140,
  marginBottom: 20,
  resizeMode: 'contain',
},
logoTablet: {
  width: 320,
  height: 200,
  marginBottom: 30,
},
text: {
  fontSize: 18,
  textAlign: 'center',
  marginBottom: 15,
  color: '#cbd5e1',
  lineHeight: 24,
},
textTablet: {
  fontSize: 22,
  textAlign: 'left',
  lineHeight: 30,
},
buttonContainer: {
  marginVertical: 20,
  width: '70%',
```

```

},
buttonContainerTablet: {
  width: '60%',
},
subHeader: {
  fontSize: 22,
  fontWeight: '600',
  color: '#f1f5f9',
  marginBottom: 10,
  marginTop: 5,
},
subHeaderTablet: {
  fontSize: 28,
},
scrollContainer: {
  maxHeight: 250,
  width: '100%',
},
scrollContainerTablet: {
  flex: 1, // take all available space
  maxHeight: undefined, // remove height restriction
},
scrollContent: {
  paddingBottom: 20,
},
scrollContentTablet: {
  paddingBottom: 40,
},
scrollItemBox: {
  backgroundColor: 'rgba(255,255,255,0.05)',
  marginVertical: 8,
  padding: 12,
  borderRadius: 10,
},
scrollItemBoxTablet: {
  padding: 16,
},
scrollItem: {
  fontSize: 16,
  color: '#e2e8f0',
},
scrollItemTablet: {
  fontSize: 20,
},

```



```
});
```

**Assessment Criteria:**

- All required components is used (40%)
- Layout is functional and readable (30%)
- App runs without errors (30%)