# Department of Computing

Muhammad Ali Sadiq

BSCS 7A

210389

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab 11: React Native Calculator Application

**Date: 05 December, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

# Lab 11: React Native Calculator Application

**Introduction**

React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.

**Objectives**

This lab will get students familiar with the React Native application Development.

**Tools/Software Requirement**

React native, Android Studio, JDK, node JS

**Description**

**Reference Videos**

<https://www.youtube.com/watch?v=TkYTPSVvMaM&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=11>

<https://www.youtube.com/watch?v=f3K2QuFH9yc&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=12>

<https://www.youtube.com/watch?v=487ec0OCppw&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=13>

<https://www.youtube.com/watch?v=8PVWlBwiegY&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=14>

<https://www.youtube.com/watch?v=4vRTFKI4ZS8&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=15>

<https://www.youtube.com/watch?v=8bhKXfEpyEw&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=16>

<https://www.youtube.com/watch?v=I-aeTW40yls&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=17>

<https://www.youtube.com/watch?v=YTkzfdyxNbM&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=18>

**Lab Task**

Create a basic calculator app in react native

|  |
| --- |
| **Solution** |
| **Task Code:**  **import React, { Component } from "react";**  **import { Platform, View, Text, ToastAndroid, ScrollView } from "react-native";**  **//Styles**  **import styles from "./styles";**  **//Custom Components**  **import NumberButtons from "./src/components/NumberButtons";**  **//constants**  **const buttons = [**  **["C", "+/-", "%", "÷"],**  **["7", "8", "9", "x"],**  **["4", "5", "6", "+"],**  **["1", "2", "3", "-"],**  **[".", "0", "Del", "="]**  **];**  **const initialOutput = "0";**  **const maxLength = 57;**  **//Serves as the Container Class**  **export default class App extends Component {**  **//Initialization**  **constructor(props) {**  **super(props);**  **this.state = {**  **\_output: initialOutput,**  **\_mathExpression: "",**  **};**  **this.\_handleEvent = this.\_handleEvent.bind(this); }**  **//Handles actions on button press**  **\_handleEvent = value => {**  **if (!isNaN(value) || value == ".") {**  **this.\_concatToOutput(value);**  **} else {**  **switch (value) {**  **case buttons[0][0]:**  **this.\_initOutput();**  **break;**  **case buttons[4][2]:**  **if (this.state.\_output.length === 1) {**  **this.\_initOutput();**  **} else {**  **this.\_replaceLastIndex("");**  **}**  **break;**  **case buttons[4][3]:**  **this.\_evaluate();**  **break;**  **case buttons[0][2]:**  **this.\_inputPercent();**  **break;**  **case buttons[0][1]:**  **this.\_toggleSign();**  **break;**  **default:**  **var strLastChar = this.state.\_output.slice(-1);**  **if (isNaN(strLastChar)) {**  **this.\_replaceLastIndex(value);**  **} else {**  **this.\_concatToOutput(value);**  **}**  **break;**  **}**  **}**  **};**  **//Function to concat user input to output screen**  **\_concatToOutput = value => {**  **if (this.state.\_output.length >= maxLength) {**  **this.\_showToast(**  **"Maximum Expression Length of " + maxLength + " is reached."**  **);**  **} else {**  **if (this.state.\_output !== initialOutput) {**  **this.setState({ \_output: this.state.\_output + "" + value + "" });**  **} else {**  **this.setState({ \_output: value + "" });**  **}**  **}**  **};**  **//Function to replace the last index of the output**  **\_replaceLastIndex = value => {**  **var str1 = this.state.\_output.replace(/.$/, value);**  **this.setState({**  **\_output: str1**  **});**  **};**  **//Validate and Calculate the output state as a Mathematical expression**  **\_evaluate = () => {**  **try {**  **let strCurOutput = this.state.\_output;**  **if (isNaN(strCurOutput)) {**  **let dEval = eval(this.\_convertToMathExpression(this.state.\_output));**  **this.setState({**  **\_output: "" + dEval,**  **});**  **}**  **} catch (exception) {**  **/\* console.log('exception: ' + exception); \*/**  **this.\_showToast("Invalid format used.");**  **}**  **};**  **\_toggleSign() {**  **let output = this.state.\_output;**  **let newValue = parseFloat(output) \* -1;**  **this.setState({**  **\_output: String(newValue)**  **});**  **}**  **\_inputPercent = () => {**  **let output = this.state.\_output;**  **let currentValue = parseFloat(output);**  **if (currentValue === 0) return;**  **const fixedDigits = output.replace(/^-?\d\*\.?/, "");**  **const newValue = parseFloat(output) / 100;**  **this.setState({**  **\_output: String(newValue.toFixed(fixedDigits.length + 2))**  **});**  **};**  **//Function to convert the output state into a valid mathematical expression**  **\_convertToMathExpression = value => {**  **let strTemp = value.replace(**  **new RegExp(this.\_escapeRegExp(buttons[0][3]), "g"),**  **"/"**  **);**  **strTemp = strTemp.replace(**  **new RegExp(this.\_escapeRegExp(buttons[1][3]), "g"),**  **"\*"**  **);**  **return strTemp;**  **};**  **\_escapeRegExp = str => {**  **return str.replace(/([.\*+?^=!:${}()|\[\]\/\\])/g, "\\$1");**  **};**  **//Function to initialize output state**  **\_initOutput = () => {**  **this.setState({**  **\_output: initialOutput**  **});**  **};**  **//Function to display an android toast**  **\_showToast = value => {**  **ToastAndroid.show(value, ToastAndroid.SHORT);**  **};**  **render() {**  **return (**  **<View style={styles.container}>**  **<View style={styles.statusBar} />**  **<View style={styles.contOutput}>**  **<View style={styles.placeHolderOutput}>**  **<Text style={styles.txtDefault}>{this.state.\_output}</Text>**  **</View>**  **</View>**  **<View style={styles.contButtons}>**  **<NumberButtons onBtnPress={this.\_handleEvent} buttons={buttons} />**  **</View>**  **</View>**  **);**  **}**  **}**  **BUTTON Component:**  **import React, { Component } from "react";**  **import {**  **View,**  **Text,**  **TouchableNativeFeedback,**  **TouchableHighlight**  **} from "react-native";**  **//Styles**  **import styles from "./styles";**  **export default class NumberButtons extends Component {**  **//This is for optimization**  **//Component should render only once**  **shouldComponentUpdate(nextProps, nextState) {**  **return false;**  **}**  **//This will call the bound function from its parent component**  **//to handle button press action/event**  **\_handleOnPress = value => {**  **requestAnimationFrame(() => {**  **this.props.onBtnPress(value);**  **});**  **};**  **render() {**  **return (**  **<View style={styles.container}>**  **{this.props.buttons.map((row, index) => (**  **<View key={index} style={styles.contRow}>**  **{row.map((col, index) => (**  **<TouchableHighlight**  **style={{ flex: 1 }}**  **key={index}**  **onPress={() => this.\_handleOnPress(col)}**  **underlayColor="#34495e"**  **>**  **<View**  **style={**  **index >= 3 ? styles.contButtonHighlight : styles.contButton**  **}**  **>**  **<Text style={styles.txtDefault}>{col}</Text>**  **</View>**  **</TouchableHighlight>**  **))}**  **</View>**  **))}**  **</View>**  **);**  **}**  **}**  **Task Output Screenshot:** |

### Deliverable

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva/quiz related to the tasks. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Ms. Ayesha Asif: [ayesha.asif@seecs.edu.pk](mailto:ayesha.asif@seecs.edu.pk).