# Department of Computing

**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:  App.js:  import React from 'react';  import { LinearGradient } from 'expo';  import { StyleSheet, StatusBar } from 'react-native';  import CalculatorResponse from './components/CalculatorResponse';  import CalculatorButtonsContainer from './components/CalculatorButtonsContainer';  export default class App extends React.Component {  constructor() {  super();  this.state = {  first: '0',  second: '',  operator: '',  result: 0,  isResult: false,  };  this.refresh = this.refresh.bind(this);  this.handleButtonPress = this.handleButtonPress.bind(this);  this.getResult = this.getResult.bind(this);  }  getResult() {  const { first, second, operator } = this.state;  const parsedFirst = parseFloat(first);  const parsedSecond = parseFloat(second) || 0;  let result = 0;  switch (operator) {  case '+':  result = parsedFirst + parsedSecond;  break;  case '−':  result = parsedFirst - parsedSecond;  break;  case '×':  result = parsedFirst \* parsedSecond;  break;  case '÷':  if (!parsedSecond || parsedSecond === 0) {  result = 'Error';  } else {  result = parseFloat(parsedFirst / parsedSecond).toFixed(8);  }  break;  default:  *// console.log('wrong operator');*  }  this.setState({  result,  isResult: true,  });  }  refresh() {  this.setState({  first: '0',  second: '',  operator: '',  result: 0,  });  }  handleButtonPress(button) {  const { isResult } = this.state;  let { first, second, operator } = this.state;  switch (button) {  case '0':  if (!isResult) {  if (!operator) {  if (first[0] !== '0' || first.length !== 1) {  first += '0';  }  } else if (second[0] !== '0' || second.length !== 1) {  second += '0';  } else {  second = '0';  }  this.setState({ first, second, operator });  } else {  this.setState({  first: '0',  second: '',  operator: '',  result: 0,  isResult: false,  });  }  break;  case '1':  case '2':  case '3':  case '4':  case '5':  case '6':  case '7':  case '8':  case '9':  if (!isResult) {  if (!operator) {  if (first[0] === '0' && first.length === 1) {  first = button;  } else {  first += button;  }  } else if (second[0] === '0' && second.length === 1) {  second = button;  } else {  second += button;  }  this.setState({ first, second, operator });  } else {  this.setState({  first: button,  second: '',  operator: '',  result: 0,  isResult: false,  });  }  break;  case '.':  if (!operator) {  if (!first.includes('.')) {  first += button;  }  } else if (!second.includes('.')) {  second += button;  }  this.setState({ first, second, operator });  break;  case '+':  case '−':  case '×':  case '÷':  if (!operator) {  operator = button;  this.setState({ first, second, operator });  } else {  this.getResult();  }  break;  case '=':  this.getResult();  break;  default:  *// console.log('wrong operator');*  }  }  render() {  const { first, second, operator, result } = this.state;  return (  <LinearGradient colors={['#3498db', '#8e44ad']} style={styles.container}>  <CalculatorResponse  first={first}  second={second}  operator={operator}  result={result}  refresh={this.refresh}  />  <CalculatorButtonsContainer  handleButtonPress={this.handleButtonPress}  />  <StatusBar barStyle="light-content" />  </LinearGradient>  );  }  }  const styles = StyleSheet.create({  container: {  flex: 1,  },  });  export default App;  CalculatorButton.js:  import React from 'react';  import { Text, StyleSheet, TouchableOpacity } from 'react-native';  class CalculatorButton extends React.Component {  render() {  const { operator, handleButtonPress } = this.props;  return (  <TouchableOpacity  style={styles.container}  onPress={() => handleButtonPress(operator)}  >  <Text style={styles.item}>  { operator }  </Text>  </TouchableOpacity>  );  }  }  const styles = StyleSheet.create({  container: {  flex: 1,  justifyContent: 'center',  alignItems: 'center',  backgroundColor: 'rgba(255, 255, 255, 0.1)',  margin: 1,  },  item: {  color: '#fff',  fontSize: 26,  },  });  export default CalculatorButton;  CalculatorButtonContainer.js:  import React from 'react';  import { View, StyleSheet } from 'react-native';  import CalculatorButton from './CalculatorButton';  class CalculatorButtonsContainer extends React.Component {  render() {  const { handleButtonPress } = this.props;  return (  <View style={styles.container}>  <View style={styles.row}>  <CalculatorButton operator={'+'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'−'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'×'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'÷'} handleButtonPress={handleButtonPress} />  </View>  <View style={styles.row}>  <CalculatorButton operator={'7'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'8'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'9'} handleButtonPress={handleButtonPress} />  </View>  <View style={styles.row}>  <CalculatorButton operator={'4'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'5'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'6'} handleButtonPress={handleButtonPress} />  </View>  <View style={styles.row}>  <CalculatorButton operator={'1'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'2'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'3'} handleButtonPress={handleButtonPress} />  </View>  <View style={styles.row}>  <CalculatorButton operator={'0'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'.'} handleButtonPress={handleButtonPress} />  <CalculatorButton operator={'='} handleButtonPress={handleButtonPress} />  </View>  </View>  );  }  }  const styles = StyleSheet.create({  container: {  flex: 1,  },  row: {  flex: 1,  flexDirection: 'row',  },  });  export default CalculatorButtonsContainer;  CalculatorResponse.js:  import React from 'react';  import { View, Text, StyleSheet, TouchableOpacity } from 'react-native';  class CalculatorResponse extends React.Component {  render() {  const { first, second, operator, result, refresh } = this.props;  const input = `${first} ${operator} ${second}`;  return (  <View>  <View style={styles.resultContainer}>  <Text style={styles.result}>  { result }  </Text>  </View>  <View style={styles.inputContainer}>  <TouchableOpacity onPress={refresh}>  <Text style={styles.delete}>  {*/\* ⏎ \*/*}  AC  </Text>  </TouchableOpacity>  <Text style={styles.input}>  { first === '0' && !operator ? 'Enter your operation' : input }  </Text>  </View>  </View>  );  }  }  const styles = StyleSheet.create({  resultContainer: {  alignItems: 'flex-end',  backgroundColor: 'rgba(255, 255, 255, 0.1)',  marginTop: 25,  paddingVertical: 25,  paddingRight: 10,  margin: 1,  },  result: {  color: '#fff',  fontSize: 42,  },  inputContainer: {  flexDirection: 'row',  justifyContent: 'space-between',  backgroundColor: 'rgba(255, 255, 255, 0)',  paddingVertical: 10,  paddingHorizontal: 5,  },  delete: {  color: 'rgba(255, 255, 255, 0.5)',  fontSize: 23,  paddingVertical: 5,  paddingHorizontal: 10,  },  input: {  color: 'rgba(255, 255, 255, 0.9)',  fontSize: 23,  padding: 5,  },  });  export default CalculatorResponse;  Task Output Screenshot:  A simple calculator App using React Native |
|  |

### 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).