# 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:  import \* as React from 'react';  import { Text, View, StyleSheet, Button, TouchableOpacity } from 'react-native';  import Constants from 'expo-constants';  *// You can import from local files*  import AssetExample from './components/AssetExample';  *// or any pure javascript modules available in npm*  import { Card } from 'react-native-paper';  export default class App extends React.Component {  constructor(){  super()  this.state = {  resultText: "",  calculationText: ""  }  this.operations = ['D', '+', '-', '\*', '/']  }  calculateResult(){  const text = this.state.resultText  this.setState({  calculationText: eval(text)  })  }  validate(){  const text = this.state.resultText  switch(text.slice(-1)){  case '+':  case '-':  case '\*':  case '/':  return false  }  return true  }  buttonPressed(text){  console.log(text)  if(text == '='){  return this.calculateResult(this.state.resultText)  }  this.setState({  resultText: this.state.resultText+text  })  }  operate(operation){  let text = ''  let lastChar  switch(operation){    case 'D':  text = this.state.resultText.split('')  text.pop()  this.setState({  resultText: text.join('')  })  break  case '+':  case '-':  case '\*':  case '/':  lastChar = this.state.resultText.split('').pop()  console.log('hi')  if(this.operations.indexOf(lastChar) > 0){  return  }  if(this.state.text== ""){  return  }  this.setState({  resultText: this.state.resultText+operation  })  }  }  render() {  let rows = []  let nums = [[1,2,3], [4,5,6], [7,8,9], ['.',0,'=']]  for(let i=0; i<4; i++){  let row = []  for(let j=0; j<3; j++){  row.push(<TouchableOpacity key={nums[i][j]} onPress={()=> this.buttonPressed(nums[i][j])} style={styles.btn}><Text style={styles.btnText}>{nums[i][j]}</Text></TouchableOpacity>)  }  rows.push(<View style={styles.row}>{row}</View>)  }  let ops = []  for(let i=0; i<5; i++){  ops.push(<TouchableOpacity onPress={()=> this.operate(this.operations[i])} style={styles.btn}><Text style={styles.btnText, styles.white}>{this.operations[i]}</Text></TouchableOpacity>)  }  return (  <View style={styles.container}>  <View style={styles.result}>  <Text style={styles.resultText}>{this.state.resultText}</Text>  </View>  <View style={styles.calculation}>  <Text style={styles.calculationText}>{this.state.calculationText}</Text>  </View>  <View style={styles.buttons}>  <View style={styles.numbers}>  {rows}  </View>  <View style={styles.operations}>  {ops}  </View>  </View>  </View>  );  }  }  const styles = StyleSheet.create({  container: {  flex: 1,  },  row:{  flexDirection: 'row',  flex:1,  justifyContent: 'space-around',  alignItems: 'center'  },  white:{  color: 'white',  fontSize: 35  },  btn:{  flex: 1,  alignItems: 'center',  alignSelf: 'stretch',  justifyContent: 'center'  },  btnText:{  fontSize: 35,  color: 'white'  },  calculationText:{  fontSize: 24,  color: 'black'  },  resultText:{  fontSize: 30,  color: 'black'  },  result:{  flex: 2,  backgroundColor: 'white',  justifyContent: 'center',  alignItems: 'flex-end'  },  calculation:{  flex: 1,  backgroundColor: 'white',  justifyContent: 'center',  alignItems: 'flex-end'  },  buttons:{  flexGrow: 7,  flexDirection: 'row'  },  numbers:{  flex: 3,  backgroundColor: '#434343',    },  operations:{  flex:1,  justifyContent: 'space-around',  alignItems: 'stretch',  backgroundColor: '#636363'  }  });  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).