# 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**

**M. Muneeb Hashmi**

**BSCS-7A**

**#220042**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

# Lab 11: React Native Calculator Application

**Lab Task**

Create a basic calculator app in react native

|  |
| --- |
| Solution |
| Task Code:  import React from 'react';  import {  StyleSheet,  View,  Text,  TouchableOpacity,  } from 'react-native';  class App extends React.Component {  constructor() {  super();  this.state = {  resultText : "",  output : ""  }  }  calculateResultulateResult() {  if (['+', '-', '/', '\*'].indexOf(this.state.resultText.slice(-1)) !== -1) return;  const expression = this.state.resultText;  this.setState({  output : eval(expression)  })  }  buttonPressed(num) {  if (num == '=') {  return this.calculateResultulateResult();  }  if (num == '.' && this.state.resultText.split('').pop() == '.') return;  this.setState({  resultText : this.state.resultText + num  })  }  operate(op) {  switch(op) {  case 'Del':  let text = this.state.resultText.split('')  text.pop()  this.setState({  resultText: text.join('')  });  break;  case '+':  if (!this.state.resultText) return;  if (['+', '-', '/', '\*'].indexOf(this.state.resultText.slice(-1)) !== -1) return;  this.setState({  resultText : this.state.resultText + '+'  });  break;  case '\*':  if (!this.state.resultText) return;  if (['+', '-', '/', '\*'].indexOf(this.state.resultText.slice(-1)) !== -1) return;  this.setState({  resultText : this.state.resultText + '\*'  });  break;  case '-':  if (!this.state.resultText) return;  if (['+', '-', '/', '\*'].indexOf(this.state.resultText.slice(-1)) !== -1) return;  this.setState({  resultText : this.state.resultText + '-'  });  break;  case '/':  if (!this.state.resultText) return;  if (['+', '-', '/', '\*'].indexOf(this.state.resultText.slice(-1)) !== -1) return;  this.setState({  resultText : this.state.resultText + '/'  });  break;  }  }  render() {  let rows = [];  let buttons = [['1', '2', '3'], ['4', '5', '6'], ['7', '8', '9'], ['.', '0', '=']];  buttons.forEach(elems => {  let row = [];  elems.forEach(elem => {  row.push(  <TouchableOpacity key={ elem } onPress={ () => this.buttonPressed(elem) } style={styles.btn}>  <Text style={styles.btnText}>  {elem}  </Text>  </TouchableOpacity>  );  });  rows.push(  <View key={ elems[0] } style={styles.row}>  {row}  </View>  );  });  let operations = ['Delete', '+', '-', '\*', '/'];  let ops = [];  operations.forEach(button => {  ops.push(  <TouchableOpacity key={ button } onPress={ () => this.operate(button) } style={styles.btn}>  <Text style={styles.btnText}>  {button}  </Text>  </TouchableOpacity>  );  });  return (  <View style={styles.container}>  <View style={styles.result}>  <Text style={styles.resultText}>{ this.state.resultText }</Text>  </View>  <View style={styles.calculateResultulation}>  <Text style={styles.calculateResultulationText}>{ this.state.output }</Text>  </View>  <View style={styles.buttons}>  <View style={styles.numbers}>  {rows}  </View>  <View style={styles.operations}>  {ops}  </View>  </View>  </View>  );  }  };  // Styles  const styles = StyleSheet.create({  container : {  flex : 1  },  row : {  flexDirection : 'row',  justifyContent: 'space-around'  },  btn : {  flex : 1,  alignItems : 'center',  justifyContent : 'center',  alignSelf : 'stretch'  },  btnText : {  color : 'white',  fontSize : 24,  alignContent : 'stretch'  },  result : {  flex : 2,  padding: 10,  backgroundColor : '#113',  justifyContent : 'center',  alignItems : 'flex-end'  },  resultText : {  color: 'white',  fontSize : 30  },  calculateResultulation : {  flex : 1,  padding: 10,  backgroundColor : '#F1692',  justifyContent : 'center',  alignItems : 'flex-end'  },  calculateResultulationText : {  color: 'white',  fontSize : 24  },  buttons : {  flex : 7,  flexDirection : 'row'  },  numbers : {  flex : 3,  backgroundColor : '#111',  justifyContent: 'space-around'  },  operations : {  flex : 1,  backgroundColor : 'black',  justifyContent: 'space-around'  }  });  export default App;  Task Output Screenshot: |