# 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**

# 

# Name: Amna Muqeem

# Class: BSCS 7B

# CMS ID: 216259

# Lab 11: React Native Calculator Application

|  |
| --- |
| Solution |
| Task Code:  import React, { Component } from "react";  import { StyleSheet, Text, View, TextInput, Button, TouchableOpacity } from "react-native";  class App extends Component {  constructor() {  super();  this.state = {  resultText: "",  calculationText: ""  };  this.operations = ["AC", "/", "\*", "-", "+"];  }  calculationResult() {  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;  }  \_onPressButton(text) {  console.log(text);  if (text == "=") {  return this.validate() && this.calculationResult(this.state.resultText);  }  this.setState({  resultText: this.state.resultText + text  });  }  operate(operation) {  switch (operation) {  case "AC":  console.log(this.state.resultText);  let text = this.state.resultText.split("");  text.pop();  this.setState({  resultText: text.join("")  });  break;  case "/":  case "\*":  case "-":  case "+":  const lastChar = this.state.resultText.split("").pop();  if (this.operations.indexOf(lastChar) > 0) return;  if (this.state.text == "") return;  this.setState({  resultText: this.state.resultText + operation  });  }  }  render() {  let rows = [];  let nums = [[7, 8, 9], [4, 5, 6], [1, 2, 3], [".", 0, "="]];  for (let i = 0; i < 4; i++) {  let row = [];  for (let j = 0; j < 3; j++) {  row.push(  <TouchableOpacity  key={nums[i][j]}  style={styles.btn}  onPress={() => this.\_onPressButton(nums[i][j])}  >  <Text style={styles.btnText}>{nums[i][j]}</Text>  </TouchableOpacity>  );  }  rows.push(  <View key={i} style={styles.row}>  {row}  </View>  );  }  let ops = [];  for (let i = 0; i < 5; i++) {  ops.push(  <TouchableOpacity  key={this.operations[i]}  style={styles.btn}  onPress={() => this.operate(this.operations[i])}  >  <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: "stretch"  },  resultText: {  fontSize: 25,  paddingRight: 10,  color: "pink"  },  btnText: {  fontSize: 40,  color: "white"  },  white: {  color: "white"  },  btn: {  flex: 1,  alignItems: "center",  alignSelf: "stretch",  justifyContent: "center"  },  result: {  flex: 2,  backgroundColor: "white",  justifyContent: "center",  alignItems: "flex-end"  },  calculation: {  flex: 1,  backgroundColor: "white",  justifyContent: "center",  alignItems: "flex-end"  },  calculationText: {  fontSize: 50,  paddingRight: 10,  color: "#bf497e"  },  buttons: {  flex: 7,  flexDirection: "row"  },  numbers: {  flex: 3,  padding: 1,  backgroundColor: "pink"  },  operations: {  flex: 1,  justifyContent: "space-around",  alignItems: "stretch",  backgroundColor: "#bf497e"  }  });  export default App;  Task Output Screenshot: |