5

Syed Muhammad Akbar (237544)

msyed.bscs17seecs

Abstract

Making a calculator app

AdvanceD programming LAB 11

REACT NATIVE

BSCS 7B

|  |
| --- |
| App.js |
| import React, { Component } from 'react';    import{  StyleSheet,  Text,  View,  TouchableOpacity  } from 'react-native';    export default class App extends Component{    constructor(){  super();  this.state = {  resultText: "",  screen: "0"  };  }    calculateResult(){  let text = this.state.resultText;  // use eval in the future  if(text[text.length -1] == '+' || text[text.length -1] == '-'){  text = text + '0';  }else if(text[text.length -1] == '\*' || text[text.length -1] == '/'){  text = text + '1';  }  this.setState({  screen: eval(text)  })  }    del(){  let text = this.state.resultText.split('');  text.pop();  this.setState({  resultText: text.join('')  })  }    ac(){    this.setState({  resultText: "",  screen: "0"  })  }    buttonPressed(text){  //text = text.toString();  if(text == '='){  return this.calculateResult();  }else if(text == 'del'){  this.del();  }else if(text == 'ac'){  this.ac();  }  else{  this.setState({  resultText: this.state.resultText + text  })  }  }    render(){  let rows = [];  for (let i = 0; i < 4; i++){  let row = []  for(let j = 0; j < 3; j++){  row.push(  <TouchableOpacity style={styles.btn}>  <Text>{i+1}</Text>  </TouchableOpacity>  );  }  rows.push(  <View style={styles.row}></View>  )  }    return(  <View style={styles.container}>  <View style={styles.result}>  <Text style={styles.resultText}>  {this.state.resultText}  </Text>  </View>  <View style={styles.calculations}>  <Text style={styles.calculationsText}>{this.state.screen}</Text>  </View>  <View style={styles.button}>  <View style={styles.numbers}>  <View style={styles.row}>  <TouchableOpacity onPress={() => this.buttonPressed(7)} style = {styles.btn}>  <Text style = {styles.btntext}>7</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(8)} style = {styles.btn}>  <Text style = {styles.btntext}>8</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(9)} style = {styles.btn}>  <Text style = {styles.btntext}>9</Text>  </TouchableOpacity>  </View>  <View style={styles.row}>  <TouchableOpacity onPress={() => this.buttonPressed(4)} style = {styles.btn}>  <Text style = {styles.btntext}>4</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(5)} style = {styles.btn}>  <Text style = {styles.btntext}>5</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(6)} style = {styles.btn}>  <Text style = {styles.btntext}>6</Text>  </TouchableOpacity>  </View>  <View style={styles.row}>  <TouchableOpacity onPress={() => this.buttonPressed(1)} style = {styles.btn}>  <Text style = {styles.btntext}>1</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(2)} style = {styles.btn}>  <Text style = {styles.btntext}>2</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(3)} style = {styles.btn}>  <Text style = {styles.btntext}>3</Text>  </TouchableOpacity>  </View>  <View style={styles.row}>  <TouchableOpacity onPress={() => this.buttonPressed('.')} style = {styles.btn}>  <Text style = {styles.btntext}>.</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed(0)} style = {styles.btn}>  <Text style = {styles.btntext}>0</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed('=')} style = {styles.btn}>  <Text style = {styles.btntext}>=</Text>  </TouchableOpacity>  </View>  </View>  <View style={styles.operations}>  <TouchableOpacity onPress={() => this.buttonPressed('ac')} style = {styles.btn}>  <Text style = {[styles.btntext, styles.white]}>A/C</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed('del')} style = {styles.btn}>  <Text style = {[styles.btntext, styles.white]}>DEL</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed('+')} style = {styles.btn}>  <Text style = {[styles.btntext, styles.white]}>+</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed('-')} style = {styles.btn}>  <Text style = {[styles.btntext, styles.white]}>-</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed('\*')} style = {styles.btn}>  <Text style = {[styles.btntext, styles.white]}>\*</Text>  </TouchableOpacity>  <TouchableOpacity onPress={() => this.buttonPressed('/')} style = {styles.btn}>  <Text style = {[styles.btntext, styles.white]}>/</Text>  </TouchableOpacity>  </View>  </View>  </View>  )  }  }    const styles = StyleSheet.create({  container: {  flex: 1  },  resultText:{  fontSize: 30,  color: 'black'  },  btn:{  flex:1,  alignItems: 'center',  alignSelf: 'stretch',  justifyContent:'center'  },  btntext:{  fontSize: 30,  color:'white'  },  white:{  color: 'white'  },  calculationsText:{  fontSize: 24,  color: 'black'  },  row:{  flexDirection: 'row',  flex: 1,  justifyContent: 'space-around',  alignItems: 'center'  },  result: {  flex: 2,  backgroundColor: 'white',  justifyContent: 'center',  alignItems: 'flex-end'  },  calculations: {  flex:1,  backgroundColor:'white',  justifyContent: 'center',  alignItems: 'flex-end'  },  button: {  flexGrow: 7,  flexDirection: 'row'  },  numbers: {  flex: 3,  backgroundColor: '#434343'  },  operations: {  flex:1,  backgroundColor:'#636363',  alignItems: 'stretch',  justifyContent: 'space-around',  }  }) |

