Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7A**

# Lab 11: React Native Calculator Application

**Submitted by: Aamna Sarosh**

**Cms id: 218953**

**Lab Task**

Create a basic calculator app in react native

|  |
| --- |
| Solution |
| Task Code:  App.js  import React, { Component } from 'react';  import { View, Text } from 'react-native';  import Style from './StyleCal';  import InputButton from './Button';  const inputButtons = [  [1, 2, 3, '/'],  [4, 5, 6, '\*'],  [7, 8, 9, '-'],  [0, 'C', '=', '+'],  ];  export default class App extends Component {  constructor(props) {  super(props);  this.initialState = {  previousInputValue: 0,  inputValue: 0,  selectedSymbol: null,  };  this.state = this.initialState;  }  render() {  return (  <View style={Style.rootContainer}>  <View style={Style.displayContainer}>  <Text style={Style.displayText}>{this.state.inputValue}</Text>  </View>  <View style={Style.inputContainer}>{this.\_renderInputButtons()}</View>  </View>  );  }  \_renderInputButtons() {  let views = inputButtons.map((row, idx) => {  let inputRow = row.map((buttonVal, columnIdx) => {  return (  <InputButton  value={buttonVal}  highlight={this.state.selectedSymbol === buttonVal}  onPress={this.\_onInputButtonPressed.bind(this, buttonVal)}  key={'butt-' + columnIdx}  />  );  });  return (  <View style={Style.inputRow} key={'row-' + idx}>  {inputRow}  </View>  );  });  return views;  }  \_onInputButtonPressed(input) {  switch (typeof input) {  case 'number':  return this.\_handleNumberInput(input);  default:  return this.\_handleStringInput(input);  }  }  \_handleNumberInput(num) {  let inputValue = this.state.inputValue \* 10 + num;  this.setState({  inputValue: inputValue,  });  }  \_handleStringInput(str) {  switch (str) {  case '/':  case '\*':  case '+':  case '-':  this.setState({  selectedSymbol: str,  previousInputValue: this.state.inputValue,  inputValue: 0,  });  break;  case '=':  let symbol = this.state.selectedSymbol,  inputValue = this.state.inputValue,  previousInputValue = this.state.previousInputValue;  if (!symbol) {  return;  }  this.setState({  previousInputValue: 0,  inputValue: eval(previousInputValue + symbol + inputValue),  selectedSymbol: null,  });  break;  case 'C':  this.setState({ inputValue: 0 });  break;  }  }  }  StyleCal.js  import { StyleSheet } from 'react-native';  var Style = StyleSheet.create({  rootContainer: {  flex: 1,  },  displayContainer: {  flex: 3,  backgroundColor: '#f3f9fb',  justifyContent: 'center',  },  displayText: {  color: '#393e46',  fontSize: 42,  fontWeight: 'bold',  textAlign: 'right',  padding: 20,  },  inputContainer: {  flex: 8,  backgroundColor: '#474f85',  },  inputButton: {  flex: 1,  alignItems: 'center',  justifyContent: 'center',  borderWidth: 2,  borderColor: '#f3f9fb',  },  inputButtonHighlighted: {  backgroundColor: '#474f95',  },  inputButtonText: {  fontSize: 32,  fontWeight: 'bold',  color: 'white',  },  inputRow: {  flex: 1,  flexDirection: 'row',  },  });  export default Style;    Button.js  import React, { Component } from 'react';  import { TouchableHighlight, Text } from 'react-native';  import Style from './StyleCal';  export default class InputButton extends Component {  render() {  return (  <TouchableHighlight  style={[  Style.inputButton,  this.props.highlight ? Style.inputButtonHighlighted : null,  ]}  underlayColor="#193441"  onPress={this.props.onPress}>  <Text style={Style.inputButtonText}>{this.props.value}</Text>  </TouchableHighlight>  );  }  }  Task Output Screenshot: |